Bringing Systems Engineering Models to Large Language Models: An Integration of OPM with an LLM for Design Assistants

Ramón María García Alarcia[1][a], Pietro Russo[2][b], Alfredo Renga[2][c] and Alessandro Golkar[1][d]

[1]*Department of Aerospace and Geodesy, Technical University of Munich, Ottobrunn, Germany*

[2]*Dipartimento di Ingegneria Industriale, Università degli Studi di Napoli Federico II, Napoli, Italy*

{*ramon.garcia-alarcia, golkar*}*@tum.de, pietro.russo12@studenti.unina.it, alfredo.renga@unina.it*

Keywords: Large Language Models, Systems Engineering, Model-Based Systems Engineering, Object-Process Methodology, Engineering Design, Design Assistant.

Abstract: Although showing remarkable zero-shot and few-shot capabilities across a wide variety of tasks, Large Language Models (LLMs) are still not mature enough for off-the-shelf use in engineering design tasks. Organizations implementing model-based systems engineering practices into their product development processes can leverage on ontologies, models, and procedures to enhance LLMs applied to engineering design tasks. We present a methodology to integrate an Object-Process Methodology model of a space system into an LLM-based spacecraft design assistant and show a performance improvement, as compared to a conventional LLM. The benchmark is evaluated through subjective expert-assessed and an objective cosine-similarity-based criteria. The results motivate additional efforts in integrating Model-Based Systems Engineering practice into LLMs as means to improve their performance and reduce shortcomings such as hallucinations and black-box, untraceable behavior.

1 INTRODUCTION

In the last five years, research and development of foundation models such as Large Language Models (LLMs) has experienced exponential growth. Today, these models display excellent capabilities in text generation and question-answering tasks. Particularly after the release of ChatGPT by OpenAI in late 2022, LLMs have emerged as a technology with the potential of transforming science, engineering and business. The current efforts in industry and academia focus on the development and commercialization of applications enabled by LLMs to support a myriad of tasks typically carried out by humans. (Myers et al., 2023)

Engineering design tasks are not extraneous to the potential impact of LLMs. Many design processes can either be supported or automated to some extent by LLM-based tools, thanks to well-established standards and procedures, and data of heritage products and services that can be ingested and learned by the models. However, there are shortcomings related to LLMs that prevent them from being able to be used, off-the-shelf, to assist in engineering design tasks. Hallucinations and incapability to perform numerical calculations are some of them (Kaddour et al., 2023), impacting negatively their reliability in these tasks.

This paper presents a methodology to increase the reliability of design assistants based on LLMs, addressing some of the aforementioned challenges, by the integration through prompting of generic system models of the system under design. In particular, we apply this methodology to a spacecraft design assistant able to produce technical requirements and specifications with a high-level space mission statement as the input (Garcia Alarcia and Golkar, 2023), with Object-Process Methodology (OPM) models. The work shows an improvement in the performance of the design assistant when compared to one without an LLM+OPM integration. Our preliminary results show a promising pathway for the integration of Model-Based Systems Engineering (MBSE) constructs, in particular, systems models, into foundation models and LLMs, for a higher degree of accuracy and trustworthiness in engineering design tasks.

[a] https://orcid.org/0000-0002-3341-2509
[b] https://orcid.org/0009-0009-6365-1226
[c] https://orcid.org/0000-0002-1236-0594
[d] https://orcid.org/0000-0001-5993-2994

2 STATE OF THE ART & RESEARCH QUESTION

Foundation models, this is, general-purpose AI models trained on large quantities of data that can perform a myriad of different tasks, have gone a long way in the last half a decade. In 2018, the introduction of the Generative Pre-trained Transformer-1 (GPT-1) by OpenAI (Radford and Narasimhan, 2018) and of the Bidirectional Encoder Representations from Transformers (BERT) by Google (Devlin et al., 2019), based on the essential contributions of the transformer architecture with self-attention mechanism (Vaswani et al., 2023) and word embeddings (Mikolov et al., 2013), were the cornerstones of a revolution that has changed the Natural Language Processing (NLP) discipline with the emergence of what nowadays is known as Large Language Models (LLMs).

Today, current state-of-the-art LLMs such as GPT-4 from OpenAI (OpenAI, 2023), Gemini from Alphabet (GeminiTeam, 2023), Llama 2 from Meta (Touvron et al., 2023), and Falcon (Almazrouei et al., 2023) or Mistral (Jiang et al., 2023) (from TII and Mistral AI, on the open-source side) are models with billions of parameters that even show zero-shot capabilities in text, picture, audio or video generation, among other modalities. Research and development is steering towards the applications side, as in the case of chatbots (in particular after the introduction of ChatGPT by OpenAI) and more broadly assistants, as the question is also posed on whether these models are the predecessors of an Artificial General Intelligence (Bubeck et al., 2023). With the current state-of-the-art LLMs, there is for many tasks no longer need for building complex systems trained from scratch, and in many cases even fine-tuning base models, as the general-purpose models show excellent capabilities with one-shot or few-shot in-context learning and prompt engineering.

However, for some other tasks, such as engineering design, this is not the case. Due to the high complexity, interdependent, and iterative nature of these processes, with also specific norms and practices being instituted in each industry, the direct application of conventional LLMs is not enough. Even though this starts to be explored, for instance by NVIDIA in the chip design industry (Liu et al., 2023), there is still a need to enhance LLMs for engineering design. In this sense, the models can leverage:

- Accumulated heritage, such as previous products and services and their data
- Ontologies of the respective industrial fields, with the important concepts and their relationships
- The standards and processes that apply
- Other typical norms and practices

Despite this, shortcomings that are still faced come in the form of hallucinations (nonsensical or unfaithful text generation (Ji et al., 2023)), lack of particularly well-structured inputs and outputs, no calculation capabilities due to their statistical nature, poor domain specialization in many fields relying on proprietary data (Ling et al., 2023), and performance limits associated to the reduced context windows of the models as well as token output limitations (Ratner et al., 2023).

The augmentation of LLMs with Knowledge Graphs (structures that capture and display information in an arranged way, including the interrelations between entities) at training, validation, and also inference time has already been explored by a series of works (Agrawal et al., 2023), as means of minimizing hallucinations. However, this is to the best of our knowledge the first work that looks at augmenting LLMs with *systems engineering models*, in particular in the frame of a design assistant.

In essence, there appears to emerge a significant opportunity to incorporate foundation models, particularly LLMs, into the engineering design discipline that is still relatively unexplored, facing some challenges that shall be researched. Additionally, the industry sectors that would benefit the most from this integration are particularly the ones following systems engineering discipline practices and standards. The way systems engineering decomposes the engineering process into smaller steps going from a high level driven by the stakeholders' needs and requirements to system requirements, a system architecture, and finally a system design, and structures the product or service life-cycle (INCOSE, 2023b), is beneficial to the structuring of an LLM-based design assistant. This applies in particular to organizations following Model-Based Systems Engineering (MBSE) practices. The system models that are built during a product's design, be it in Systems Modeling Language (SysML) (Hause, 2006), Object-Process Methodology (OPM) (Dori, 2002), Unified Modeling Language (UML) (Koç et al., 2021), in code, etc., can be integrated into the LLM to improve its performance on new designs.

Our contribution answers the question of whether the integration of MBSE into an LLM, in particular of an OPM semantic model of a generic system through prompting, effectively improves the reliability of a design assistant tool based on an LLM.

3 METHODOLOGY

In this work, we integrate a generic model of the system under design, through prompting, with a design assistant based on an LLM. In particular, the design assistant first takes a space mission statement as user input in natural language. In the background, without user intervention, the generic system model is included as well into the prompts. The LLM component of the design assistant transforms that initial mission statement into a list of technical specifications.

The results of this process are then compared with the output of the same LLM without the system model integrated, to understand if the design assistant tool performances are improved by the MBSE-LLM integration. In both this and the former case, the same list of subsystem requirements or technical specifications are expected as an output, starting from the same mission statement.

3.1 Model-Based Systems Engineering Tool

The Model-Based Systems Engineering (MBSE) tool used in this work is the *Object-Process Methodology*, a conceptual language for systems modeling, which natively exploits a bi-modal representation: Object-Process Diagram (OPD), a visual representation of the system including the entities of the model (objects, processes, and states) and links and relations among them, and Object-Process Language (OPL), a list of English sentences describing the model in a human-oriented language. Each OPD, except for the System Diagram (SD), the top-level OPD, can be obtained by refinement, in-zooming, or unfolding, of a thing (object or process) in its ancestor OPD, to avoid the loss of details and to keep the overall view of the system. The text is automatically generated by the OPM software (OPCloud or OPCAT) as the model is created. (Dori, 2002) The natural language representation is ideal as it makes the integration of this MBSE tool with Large Language Models (LLMs) easier. Other MBSE tools exist, like SysML, that allow to model different aspects of a system (e.g. SysML uses four types of diagrams: structure, behavior, requirements, and parametric). However, they are practically equivalent for our purposes, since only the architecture of a generic system must be modeled. OPM has been selected because it is more intuitive and natively exploits the bi-modal representation (OPD and OPL).

Our implementation of the generic space system of systems for the integration in the design assistant is composed of three systems: space segment, ground segment, and user segment (as represented by Figure 1). The space segment includes the launcher and one or more spacecraft with their relevant subsystems; the ground segment comprises the ground stations and the mission control center; the user segment refers to the users and their services. For each subsystem, components and properties are defined. The measurable properties are indicated with the proper measurement unit. If the number of components is more than one, the properties values are defined as vectors.

As for the spacecraft, the OPM model describes its payload, the orbit, the associated ground stations for data processing, Telemetry & Telecommand subsystem, Command & Data Handling subsystem, Attitude & Orbit Control subsystem, Propulsion subsystem, Electrical & Power subsystem, Thermal Control subsystem, the structure, as well as general spacecraft parameters and the launcher to be used for orbit insertion.

The space system model has been designed to be meaningful while being as generic as possible, including a large set of potential spacecraft components. The values of the properties are all preset as *tbd* (to be defined) as they are aimed to be completed by the design assistant tool upon the generation of the technical specifications. A view of the model at a lower decomposition level is illustrated in Figure 2.

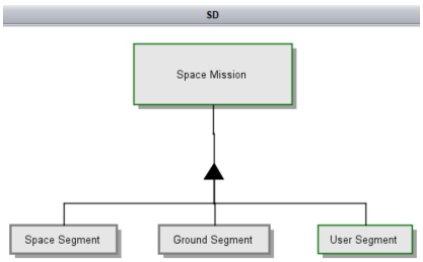


Figure 1: Object-Process Methodology System diagram of the space system of systems. Highest level view.

3.2 Large Language Model

A Large Language Model (LLM) is used to convert the high-level mission statement into technical specifications, which are used to derive the design elements necessary for the following design steps (e.g., system budgets, CAD models, etc.). Thanks to its capability of processing and generating human-like language, the LLM takes as an input the mission statement together with the generic OPL to be modified and completed. It then produces a new OPL, and so with it the list of subsystem requirements, for the specific mission of interest. The research challenge is to transform a high-level mission statement, which can be ambiguous, inconsistent, or incomplete due to the use

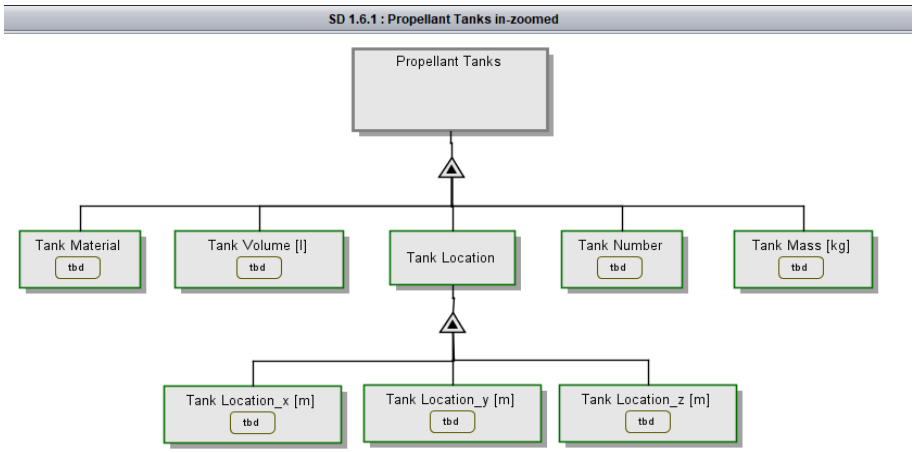


Figure 2: Object-Process Methodology diagram of a propellant tank.

of natural language, into clear, unambiguous, traceable, and comprehensive technical requirements, following systems engineering practice. At the time of the paper, many LLMs are available (GPT 3.5, GPT 4, BARD, Falcon, PaLM, Claude, and others). Each one of them has different characteristics, advantages and drawbacks. We used GPT 4 - an API based LLM - for the work presented in this paper, to facilitate the integration with other applications, e.g. a chatbot design assistant, and for its context length.

3.3 LLM + OPM Integration

The OPL is integrated into the design assistant by prompting the LLM, and one of the challenges faced is that the current OPL length (around 9600 tokens) is longer than most of the current LLMs context windows limits. Indeed, because of the direct relation between computational costs and context window size, the number of tokens an LLM can process is limited. In this work, OpenAI's GPT 4-128k (November 2023 version), codenamed *gpt-4-1106-preview*, has been used due to its larger 128k context window, through API calls. However, the OPL has been divided into smaller chunks, following the system model structure depicted in Table 1.

To control the results for the variation of the selected LLM, the objective criterion is also obtained using a different model from OpenAI, GPT 3.5, in its version *gpt-3.5-turbo-16k* with a 16k-token context window. Due to the size of the OPL and the sliding context window, it is feasible that some parts of the OPL are cut, however, this impact is reduced by the use of the average cosine similarity metric. Using other LLMs is at this point not feasible due to the small context window sizes (4096 tokens for Llama 2, 2048 tokens for Falcon, or 4096 tokens for Mistral, for instance).

Table 1: Number of tokens for each of the OPL chunks.

OPL CHUNKS	NUMBER OF TOKENS
Payload	347
Orbit	182
Ground Station	579
TT&C	766
CDH	493
AOCS	3043
Propulsion Subsystem	1071
EPS	1488
TCS	1413
Spacecraft	239
Launcher	12

These OPL pieces are given as input to the LLM step by step, by adding new inputs to the previous context, for three reasons:

- The previous input-output pairs represent a context and contain the required information so that the LLM can proceed in the design process.
- The previous input-output pairs represent an example on how to generate technical specifications, increasing the average quality of the output.
- Compatibility with LLMs having shorter context windows.

3.4 Benchmarking

To evaluate the design assistant tool's performance, two different criteria have been implemented to score the output: a subjective criterion, based on an expert's review, and an objective criterion, thought of as a numerical scoring.

3.4.1 Subjective Criterion

A tool generating technical specifications for the design of a system shall be trustworthy above all. Trustworthiness is evaluated as the weighted average of six parameters that should characterize a requirement (the output of the tool), as stated by the International Council of Systems Engineering's *Systems Engineering Body of Knowledge (SEBoK)* (INCOSE, 2023a). We propose a parametric definition of trustworthiness, decomposed as follows:

- Clarity: the output is understandable.
- Coherency: the output is compliant with the mission statement and the previous information.
- Unambiguity: the output is not subject to different interpretations and no alternatives or redundancies are generated.
- Completeness: the output is complete and sufficient to proceed to the next step of the design process.
- Traceability: the output can be traced back to higher level specifications (e.g. mission statement) and its source.
- Verifiability: the output can be easily verified through various verification processes.

Each metric receives a score, ranging from 0 to 100 %, based on the ratio between the number of output lines fulfilling the parameter and the total number of output lines. Thus, the score is the percentage of output satisfying the metric, as rated by an expert individual. Only for completeness, the overall score is the weighted average of two additional contributions: coverage (weight=0.4), representing the percentage of system aspects covered by the output; and granularity level (weight=0.6), related to the level of detail of the generated content, which must be the one of a technical specification. Coverage and granularity are assigned a 40/60 weight allocation to represent the idea that the latter is more important due to the need to obtain a sufficiently detailed list of requirements as an output to the model.

A weight is assigned to each parameter as specified by Table 2, coming from analyses, past knowledge, and experience. All the metrics contribute in the same way to trustworthiness, except for traceability, which is the most important parameter, since the source of the output plays a very important role. Indeed, tracing requirements from their origins through the system design is crucial for ensuring that the system meets its intended objectives and for managing changes effectively, making expert analyses and interventions easier. A preliminary sensitivity analysis has been performed to select and support the weights of the subjective criterion. The results are briefly described in Section 4.

Table 2: Evaluation metrics and relevant weights for the subjective criterion.

METRICS	WEIGHTS
Clarity	0.1
Coherency	0.1
Unambiguity	0.1
Completeness	0.1
Traceability	0.5
Verifiability	0.1

3.4.2 Objective Criterion

A clear problem when evaluating the output of a Large Language Model, which appears in a more accentuated manner when this output is a technical requirement or specification, is that there is not a single solution, but rather a wide variety of alternative solutions that can be properly formulated and be technically feasible. And even more, each of these valid solutions has an almost infinite amount of ways of being formulated linguistically (i.e., in text) into one or more sentences. This issue rules out the classical approaches used for evaluating Machine Learning models' output, in which the result from a model is evaluated with a validation dataset containing for each entry a set of finite correct outputs, also known as labels.

In this evaluation, we propose the use of the cosine similarity metric in order to provide an objective result that does not rely on subjective scoring. Cosine similarity is a distance-based similar metric already used for LLM output evaluation by other works in the field (Chen et al., 2023).

Cosine similarity for a vector of words $\vec{o}$ containing the words produced by the LLM, and a vector of words $\vec{g}$ containing the words of a gold standard answer -the comparison reference-, is defined as follows:

$$cos_sim(\vec{o},\vec{g}) = \frac{\vec{o}\cdot\vec{g}}{|\vec{o}||\vec{g}|} \quad (1)$$

In practical terms, we implement the cosine similarity metric in Python programming language, through *scikit-learn* library's sklearn.metrics.pairwise.cosine_similarity function. To achieve a higher degree of text format agnosticism, we remove the set of English stop words from both the output vector and the gold standard vector as specified in the *nltk* library's corpus. The words are converted from strings to word embeddings with the *gensim* library's gensim.models.word2vec function implementing the well-known word2vec algorithm

(Mikolov et al., 2013). To account for the differences in the lengths of the vectors, we use padding. In our case, we zero-pad the shortest vector to match the length of the longest one. The Appendix includes a flowchart of the objective criterion's steps, for a sinthesized view of it.

All in all, we measure how close, conceptually, is the output of the design assistant to a gold standard reference. When the output is close to the reference the score tends to 1, when it is opposite to the reference the score tends to -1. Thus, we account for different plausible possibilities in the output, as they will all be conceptually close to the gold standard. By using cosine similarity, and even more by having eliminated stop words, we also avoid formatting-induced scoring changes that other methodologies might suffer from.

4 RESULTS & DISCUSSION

4.1 Case Study

To evaluate the design assistant tool, we have chosen three space missions for which a large amount of information is publicly available, especially in terms of design choices explanations, and requirements. Comparing the results of the design assistant tool with the real design of a system helps in validation and evaluation. Having three different space missions also allows us to control the results for the variability of the mission statement. The mission statements are entered to a design assistant with an off-the-shelf LLM, and to a design assistant with an LLM that is also fed the generic OPM of the space system. Technical specifications are retrieved and assessed using the subjective criterion and the objective criterion. The first selected baseline mission is the Ten-Koh small satellite in Low Earth Orbit, developed by the Kyushu Institute of Technology (Fajardo et al., 2019). The second selected mission is the big MetOp-C meteorological satellite (4300 kg) from the European Space Agency and EUTMETSAT, flying on an 884-km Sun-Synchronous Orbit (SSO) (Righetti et al., 2020). The third baseline mission is the LunaH-Map from NASA, made of 6U CubeSats for lunar mapping (Hardgrove et al., 2019). The inputs are available below for reproducibility.

"Space radiation poses challenges to satellites, causing anomalies like single event effects, ionizing radiation-induced component degradation, and charging issues. Designing satellites capable of withstanding these anomalies requires a deep understanding of the radiation environment. Galactic cosmic rays, solar energetic particles, and trapped high-energy particles contribute to this environment, with unpredictable energy variability in the low-Earth orbit (LEO) region, where most satellites are located. High-energy electrons, protons, and ions impact spacecraft differently based on mission design, epoch, and class. Manned missions must consider particle population unpredictability in mission duration and life support systems. Recent missions like Van Allen Probes, THEMIS, MMS, ERG (ARASE), Proba-2, and Swarm have explored near-Earth space, providing direct measurements of charged particles. Create a small satellite able to characterize the plasma environment, detect MeV-range electrons, and study material sample changes in the space environment to demonstrate that small, low-cost spacecraft can offer a cost-effective approach to space environment research, utilizing commercial components. A detailed design is required."

Mission Statement 1: Ten-Koh mission

"Considering the critical role satellites play in monitoring and understanding our planet's climate and environmental changes, our goal is to contribute to the next generation of Earth observation satellites. Our envisioned satellite will be equipped with cutting-edge instrumentation designed to provide accurate and comprehensive data on Earth's atmosphere, weather patterns, and climate dynamics. We aim to enhance our understanding of key environmental indicators, including greenhouse gas concentrations, atmospheric composition, and surface temperature variations. The mission objectives include achieving a high level of precision in data collection and analysis, allowing for improved weather forecasting, climate modeling, and environmental monitoring. We aspire to contribute to global efforts in mitigating the impacts of climate change by providing policymakers, scientists, and the public with invaluable insights into Earth's dynamic systems. In the spirit of collaboration, our mission seeks to establish international partnerships to maximize the impact and reach of our satellite's observations. By fostering cooperation with other space agencies, research institutions, and commercial entities, we aim to create a robust network of data-sharing and collaborative initiatives that transcend borders. Furthermore, our satellite will be designed with scalability and adaptability in mind, ensuring that it can accommodate future technological advancements and evolving scientific requirements. This adaptability will enable our mission to remain at the forefront of Earth observation, continuously contributing to the collective knowledge about our planet's changing environment. A detailed design is required."

Mission Statement 2: MetOp-C mission

"In response to the growing interest in lunar exploration and the imminent expansion of lunar activities within the next two decades, our goal is to advance scientific knowledge and contribute to the sustainable development of lunar resources. Our focus lies in mapping the abundance of hydrogen down to one meter beneath the surface of the lunar south pole. Inspired by the renewed interest in Moon exploration and the establishment of a lunar space economy, our mission is dedicated to the development of a small satellite (CubeSat 6U) capable of providing a high-resolution map of the abundance and distribution of hydrogen-rich compounds, like water, in this region of the Moon and expand on the less accurate maps made by previous missions. Our vision is to create a cost-effective and innovative approach to lunar environment research, addressing unique challenges in the development of a CubeSat intended to run longer and travel further than most LEO CubeSat missions. LunaH-Map aims to demonstrate the capabilities of small spacecraft in conducting sophisticated space environment studies. Through LunaH-Map, we aim to pave the way for a new era of lunar exploration, where comprehensive data on water-ice and hydrogen distribution at the lunar south pole serves as a foundation for informed decision-making and sustainable resource utilization. By pushing the boundaries of technology and embracing a cost-effective approach, LunaH-Map aspires to inspire future missions and stakeholders to join us in unlocking the full potential of the Moon for the benefit of humanity and the advancement of space exploration. A detailed design of the mission is required."

Mission Statement 3: LunaH-Map mission

4.2 Results of the Subjective Criterion

The outputs are not reproduced in this article due to their lengthiness, but the results of the subjective criterion are shown in Table 3 for an off-the-shelf LLM (GPT-4 128K) and in Table 4 for the LLM+OPM integration. In all the cases the outputs are the design of the required space mission, but with different characteristics highlighted by the parameters scores. They are clear and understandable (clarity score 100%) and respect the mission statements and what is asked in input. In addition, the design choices are coherent with each other (coherency score 100%). An important difference exists, instead, in terms of ambiguity: the outputs of the LLM not integrated with an OPM model are ambiguous (unambiguity score is lower than 40%), because most of the suggested solutions can be interpreted differently and many alternatives are provided (e.g. aluminum or composite frame are possible choices for Ten-Koh structure; a propulsion system respecting mass and volume constraints is proposed without further specifications for MetOp-C and LunaH-Map), unlike the outputs of the tool which follow the structure of the system model (unambiguity score 100% for Ten-Koh and MetOp-C missions). The only exception is represented by the lunar mission (unambiguity score 85%), whose output is slightly ambiguous, because orbit, ground station and communication subsystem are not completely designed by the tool. Regarding completeness, coverage and detail level are analyzed: all the system aspects are covered by the LLM+OPM tool, while this is not always true for the LLM (MetOp-C orbit and LunaH-Map Command&Data Handling subsystem are not included in the outputs); and only the MBSE-LLM integration allows us to reach the detail level of technical specifications. The main prevalent problem is the low traceability (around 30% for the LLM and 35% for the tool). The outputs can be traced back to the mission statement, but not to their origin. Integrating the OPM with the LLM makes the requirements slightly more traceable because all the design choices belong to the generic system model. However, this is not enough to solve the problem, especially for numbers, which are the result of a probabilistic approach, instead of the application of equations and physics models. Nonetheless, the tool outputs are still verifiable (verifiability score higher than 80%), because specific solutions are provided with relevant numbers, while only small parts of the LLM outputs can be verified, due to their ambiguities and low detail level (verifiability score lower than 50%). In general, it is possible to notice that the scores of the two Earth missions are very similar, unlike the lunar mission score. The main reason can be found in the large amount of data available for space missions around the Earth with respect to interplanetary missions, which makes easier the design of an Earth-based mission. As a result of these considerations, we can state that LLM trustworthiness is very low, but it is improved by almost 50% thanks to the integration with an MBSE tool.

Based on the scores of the metrics, a preliminary sensitivity analysis has been carried out to support the choice of metric weights and thus to verify the robustness of the evaluation criterion. It consists of varying one metric weight per time, from 0 to 1, keeping all the other weights equal to each other. The results, not fully reproduced in the article for a matter of brevity, consistently show an improvement in the performance with LLM+OPM integration across all scenarios of metric weights. Moreover, traceability is the parameter with the largest impact on the LLM+OPM tool performance, as expected, while for the off-the-shelf LLM, clarity, coherency, and unambiguity are also not negligible. Figures 3 and 4 exhibit the results of the sensitivity analysis for the traceability weight in both cases.

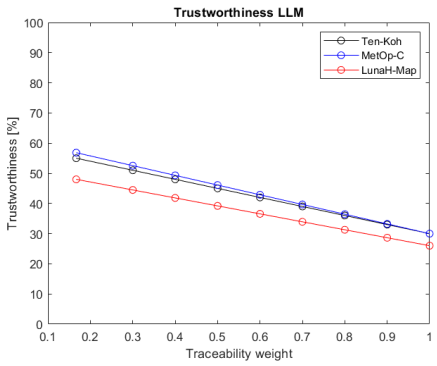


Figure 3: Effect of traceability weight variation on LLM trustworthiness.

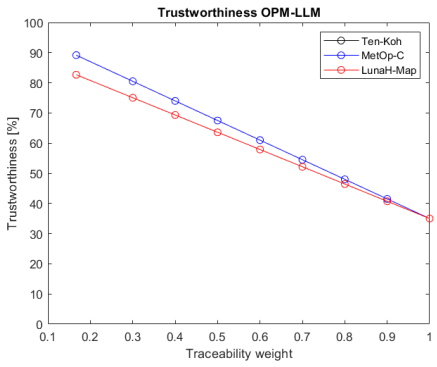


Figure 4: Effect of traceability weight variation on LLM+OPM trustworthiness.

Table 3: Off-the-shelf GPT-4 subjective evaluation results.

LLM subjective evaluation results	**Ten-Koh**	**MetOp-C**	**LunaH-Map**
Clarity	100%	100%	100%
Coherency	100%	100%	100%
Unambiguity	20%	35%	11%
Completeness (0.4*Coverage + 0.6*DetailLevel)	46% (0.4*100% + 0.6*10%)	38% (0.4*91% + 0.6*3%)	36% (0.4*91% + 0.6*0%)
Traceability	30%	30%	26%
Verifiability	35%	38%	15%
Trustworthiness	45.1%	46%	39.2%

Table 4: LLM+OPM integration subjective evaluation results.

Tool subjective evaluation results	**Ten-Koh**	**MetOp-C**	**LunaH-Map**
Clarity	100%	100%	100%
Coherency	100%	100%	100%
Unambiguity	100%	100%	85%
Completeness (0.4*Coverage + 0.6*DetailLevel)	100% (0.4*100% + 0.6*100%)	100% (0.4*100% + 0.6*100%)	91% (0.4*100% + 0.6*85%)
Traceability	35%	35%	35%
Verifiability	100%	100%	85%
Trustworthiness	67.5%	67.5%	63.6%

Table 5: Results of the objective criterion evaluation.

		Minimum		**Maximum**		**Average**		**Std. deviation**	
		LLM	**+OPM**	**LLM**	**+OPM**	**LLM**	**+OPM**	**LLM**	**+OPM**
Mission 1 (Ten-Koh)	**gpt-3.5**	0.065	0.038	1.000	1.000	0.986	0.990	0.038	0.036
	gpt-4	-0.458	0.172	1.000	1.000	0.977	0.991	0.066	0.031
Mission 2 (MetOp-C)	**gpt-3.5**	-0.582	-0.544	1.000	1.000	0.984	0.992	0.052	0.040
	gpt-4	-0.817	-0.798	1.000	1.000	0.973	0.986	0.087	0.048
Mission 3 (LunaH-Map)	**gpt-3.5**	-0.680	-0.669	1.000	1.000	0.985	0.988	0.076	0.075
	gpt-4	-0.696	-0.255	1.000	1.000	0.976	0.988	0.083	0.049

4.3 Results of the Objective Criterion

The results of the objective criterion are shown in Table 5. They are controlled for mission statement variability, with the three selected baseline missions presented before, and for the LLM variability, using *gpt-3.5-16k-turbo* and *gpt-4-1106-preview*. For each token of the output, its cosine similarity to the corresponding token in the gold standard reference is calculated, and a vector of cosine similarities is built. The table displays the minimum value encountered in that vector, the maximum, the average value, and the standard deviation. The results are presented for the design assistant with an off-the-shelf LLM without any OPM integration (labeled LLM in the table) and for the design assistant with an LLM+OPM integration (labeled +OPM in the table).

The results are clear and confirm what was seen already with the subjective criterion. They show that our OPM integration improves the quality of the design assistant's output consistently for all the mission statements and all the LLMs, as displayed in the Average columns. This is due to the space systems ontology that the OPM of a generic space mission provides, and the in-context learning thereby performed by the LLM at inference time, improving domain-specific knowledge and reducing hallucinations. Additionally, the integration of the OPM consistently improves the predictability of the results, as seen in the Standard deviation columns. This is due to the structuring of the input and the output that naturally happens when integrating a systems engineering model (in this case, the OPM), bringing a structure with a set of rules, with the LLM.

5 CONCLUSION

In this work, we have discussed the shortcomings of using off-the-shelf LLMs for engineering design tasks, particularly in the context of a design assistant for spacecraft that drafts technical requirements with a high-level mission statement as input. We have introduced a methodology for integrating Model-Based Systems Engineering models, in particular an Object-Process Methodology model, to an LLM to improve its reliability. We presented partial, preliminary results with both a subjective criterion, looking for the traits that make design assistant requirements trustworthy, and an objective criterion comparing the outputs directly to a golden reference. The results show an improvement in the quality of the outputs with the LLM+MBSE integration. These improvements are associated with the introduction of an ontology, being learned in context by the LLM, that reduces hallucinations, as well as a higher degree of structure impacting both the input and the output, all coming naturally from the properties of the system model being introduced.

In future work, we aim to increase the completeness of the results in particular with a thorough sensitivity analysis of weights of the subjective criterion. Additionally, dealing with smaller LLMs and fitting the system models to their reduced context window remains one of the biggest open challenges, in particular, to enable these kinds of design assistants to run on smaller devices in the edge. Related to it, understanding whether a deeper-level integration of system models would be more useful and reliable than prompting is also an open area for further investigation. Additionally, Large Language Models that have been trained or fine-tuned in systems engineering or more broadly engineering design data such as books, papers, and standards shall be created and thoroughly assessed to understand the performance improvement compared to the state-of-the-art generalist models.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to Prof. Dr. Dov Dori and Dr. Hanan Kohen for granting access to OPCloud for research purposes. We are also grateful to Prof. Dr. Merouane Debbah for his mentoring and insightful comments.

REFERENCES

Agrawal, G., Kumarage, T., Alghami, Z., and Liu, H. (2023). Can Knowledge Graphs Reduce Hallucinations in LLMs? : A Survey. *ArXiv*, abs/2311.07914.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., and Penedo, G. (2023). The Falcon Series of Open Language Models. *ArXiv*, abs/2311.16867.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4. *ArXiv*, abs/2303.12712.

Chen, Z., Du, W., Zhang, W., Liu, K., Liu, J., Zheng, M., Zhuo, J., Zhang, S., Lin, D., Chen, K., and Zhao, F. (2023). T-eval: Evaluating the tool utilization capability step by step. *ArXiv*, abs/2312.14033.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dori, D. (2002). *Object-Process Methodology – A Holistic Systems Paradigm*. Springer, New York.

Fajardo, I., Lidtke, A. A., Bendoukha, S. A., Gonzalez-Llorente, J., Rodríguez, R., Morales, R., Faizullin, D., Matsuoka, M., Urakami, N., Kawauchi, R., Miyazaki, M., Yamagata, N., Hatanaka, K., Abdullah, F., Rojas, J. J., Keshk, M. E., Cosmas, K., Ulambayar, T., Saganti, P., Holland, D., Dachev, T., Tuttle, S., Dudziak, R., and ichi Okuyama, K. (2019). Design, Implementation, and Operation of a Small Satellite Mission to Explore the Space Weather Effects in LEO. *Aerospace*, 6(10).

Garcia Alarcia, R. M. and Golkar, A. (2023). Architecture of a generative design tool for spacecraft and user front-end implementation through a chatbot smart design assistant. In *IAC 2023 Congress Proceedings, 74th International Astronautical Congress*.

GeminiTeam (2023). Gemini: A Family of Highly Capable Multimodal Models. *ArXiv*, abs/2312.11805.

Hardgrove, C., DuBois, J., Heffern, L., Cisneros, E., Bell, J., Crain, T., Starr, R., Prettyman, T., Lazbin, I., Roebuck, B., Struebel, N., Clark, E., Nelson, D., Bauman, J., Williams, B., Johnson, E., Christian, J., Stoddard, G., Tsay, M., Model, J., Hruby, P., Babuscia, A., Stem, S., Sanders, D., Hegel, E., Wiens, M., Parlapiano, S., Hailey, P., O'Brien, T., Mesick, K., and Coupland, D. (2019). The Lunar Polar Hydrogen Mapper (LunaH-Map) Mission. In *33rd Annual AIAA/USU Conference on Small Satellites*.

Hause, M. (2006). The SysML Modelling Language. In *Fifth European Systems Engineering Conference*.

INCOSE (2023a). *Guide to the Systems Engineering Body of Knowledge*. International Council on Systems Engineering, San Diego, CA, 2.9 edition.

INCOSE (2023b). *INCOSE systems engineering handbook*. John Wiley & Sons, Nashville, TN, 5 edition.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12).

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7B. *ArXiv*, abs/2310.06825.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. (2023). Challenges and Applications of Large Language Models. *ArXiv*, abs/2307.10169.

Koç, H., Erdoğan, A. M., Barjakly, Y., and Peker, S. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *Proceedings*, 74(1).

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., Zhao, T., Panalkar, A., Cheng, W., Wang, H., Liu, Y., Chen, Z., Chen, H., White, C., Gu, Q., Pei, J., and Zhao, L. (2023). Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey. *ArXiv*, abs/2305.18703.

Liu, M., Ene, T.-D., Kirby, R., Cheng, C., Pinckney, N., Liang, R., Alben, J., Anand, H., Banerjee, S., Bayraktaroglu, I., Bhaskaran, B., Catanzaro, B., Chaudhuri, A., Clay, S., Dally, B., Dang, L., Deshpande, P., Dhodhi, S., Halepete, S., Hill, E., Hu, J., Jain, S., Khailany, B., Kokai, G., Kunal, K., Li, X., Lind, C., Liu, H., Oberman, S., Omar, S., Pratty, S., Raiman, J., Sarkar, A., Shao, Z., Sun, H., Suthar, P. P., Tej, V., Turner, W., Xu, K., and Ren, H. (2023). ChipNeMo: Domain-Adapted LLMs for Chip Design. *ArXiv*, abs/2311.00176.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Myers, D., Mohawesh, R., Chellaboina, V. I., Sathvik, A. L., Venkatesh, P., Ho, Y.-H., Henshaw, H., Alhawawreh, M., Berdik, D., and Jararweh, Y. (2023). Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing*.

OpenAI (2023). GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Radford, A. and Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training. Preprint.

Ratner, N., Levine, Y., Belinkov, Y., Ram, O., Magar, I., Abend, O., Karpas, E., Shashua, A., Leyton-Brown, K., and Shoham, Y. (2023). Parallel context windows for large language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Righetti, P., de Juana Gamo, J., and Sancho, F. (2020). Metop-c deployment and start of three-satellite operations. *The Aeronautical Journal*, 124(1276):902–916.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv*, abs/2307.09288.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*. Association for Computing Machinery.

APPENDIX

To better illustrate the methodology's objective criterion, Figure 5 presents a flowchart of the steps followed to compute the cosine similarity values between the design assistant's output using an off-the-shelf LLM and the golden standard, as well as between the design assistant's output leveraging an LLM+OPM integration and the golden standard. The flowchart also illustrates the metrics that are calculated to compare the cosine similarity values and benchmark the improvement attained by the integration of the space mission's OPM system model.

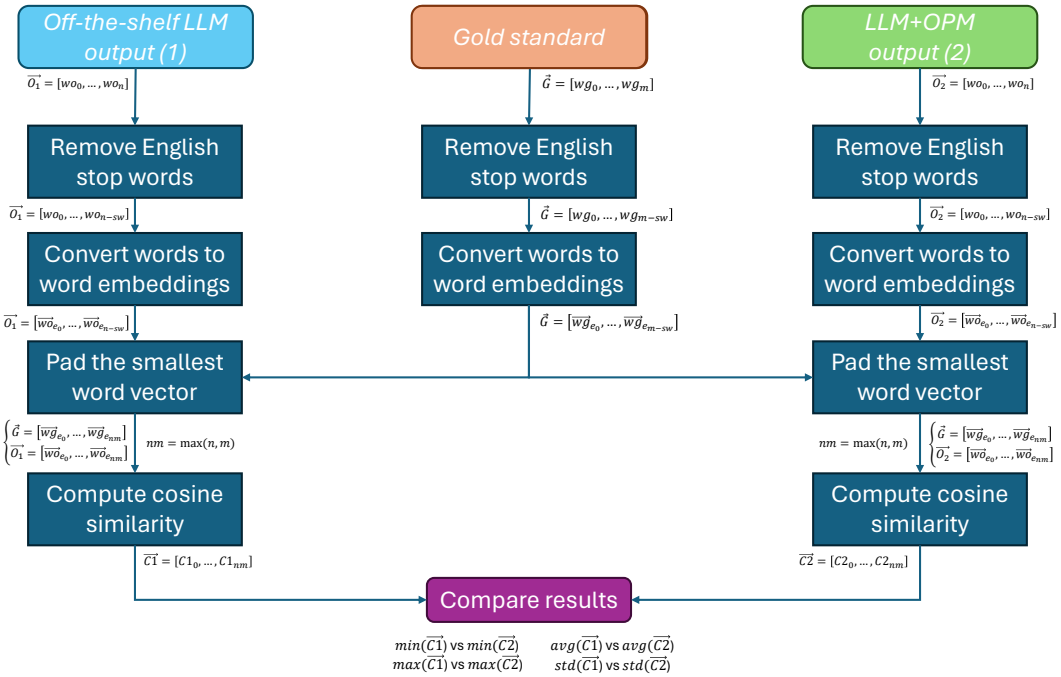


Figure 5: Flowchart of the methodology's objective criterion.